



# MIM

## Verbindt en ontzorgt de informatieoverheid



Verkent, verbindt, verankert

Zoek

[Geo-standaarden](#) [Thema's](#) [Over Geonovum](#)

[Home](#) / [Geo-standaarden](#) / [Metamodel Informatiemodellering...](#)

### Metamodel Informatiemodellering (MIM)

Om alle informatiemodellen in Nederland nog beter op elkaar aan te laten sluiten hebben VNG Realisatie, Kadaster en Geonovum gezamenlijk een metamodel ontwikkeld voor informatiemodellering. Hierin komt de modelleringskennis van deze drie organisaties samen.

→ [Bekijk alle Metamodel Informatiemodellering \(MIM\) standaarden](#)

Informatie is een motor onder het functioneren van de overheid in Nederland. In het kader van onderlinge samenwerking, wisselen overheidsorganisaties veel informatie uit. Daarbij is het van groot belang dat we hetzelfde verstaan onder de gegevens die we gebruiken en dat we gemeenschappelijke afspraken maken over hoe we gegevens beschrijven en uitwisselen.



## Voorstellen

Lennart van Bergen (Kadaster) – informatie architect, auteur  
metamodel voor informatiemodellen, werkgroep MIM

Thies Mesdag (Kadaster) – adviseur datamanagement en  
beheerder IMKAD, werkgroep MIM

# MIM factsheet



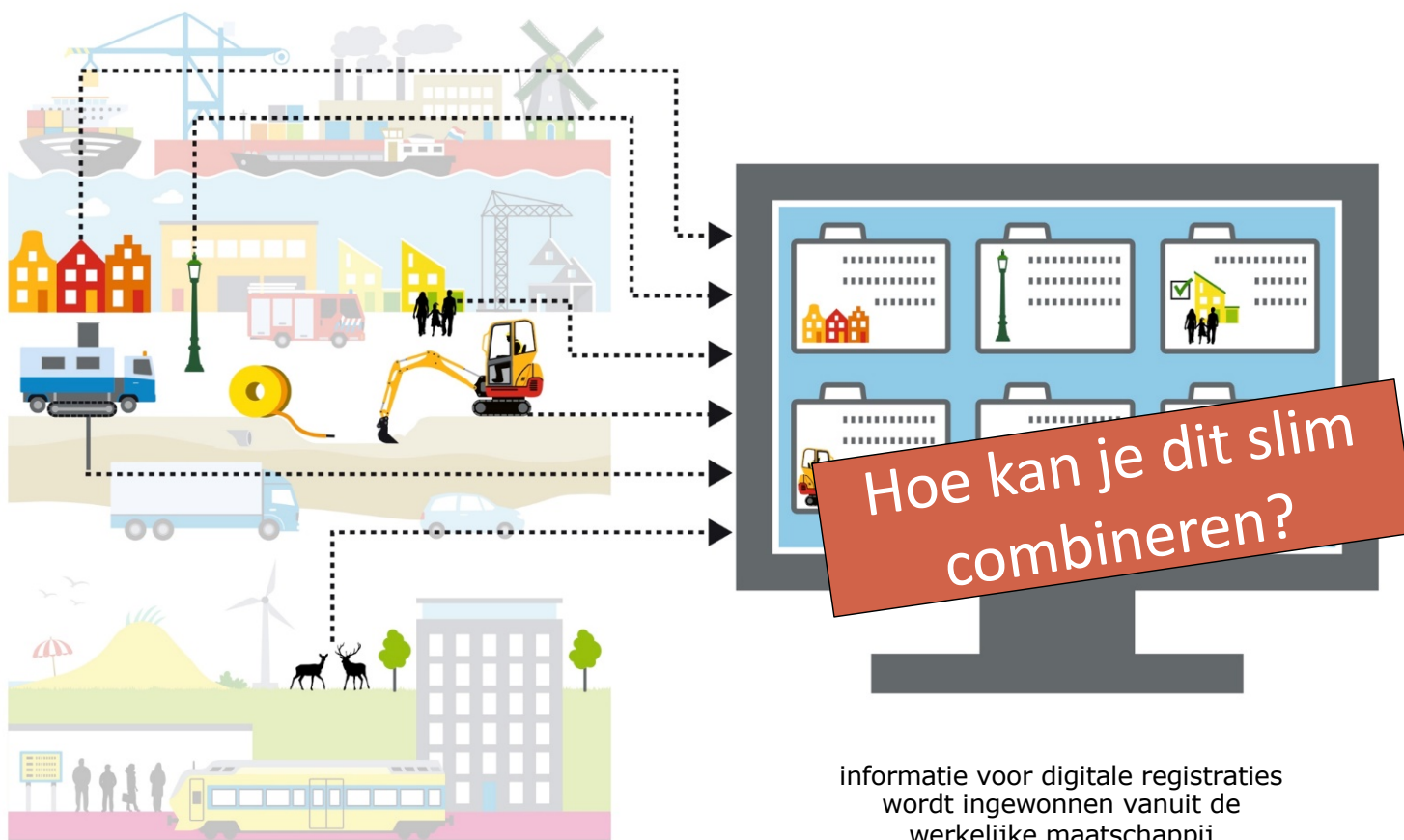
- **Naam:** MIM-Metamodel voor Informatiemodellering
- **Toepassingsgebied:** Informatiemodellen voor formaliseren van inwinning, registratie en ontsluiting van digitale gegevens
- **Waarom:** Eenduidige aanpak zorgt voor semantische integratie, harmonisatie, effectieve informatiehuishouding
- **Locatie:** <https://www.geonovum.nl/geo-standaarden/metamodel-informatiemodellering-mim/metamodel-voor-informatiemodellering-mim>
- **Ontwikkelaars:** VNG-Realisatie, Kadaster, Geonovum
- **Beheer:** Geonovum



**Werkelijkheid**



**Registraties**



werkelijkheid



Informatiemodel  
digitaal model

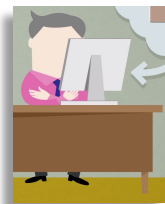


een informatiemodel is een digitaal  
model van de werkelijkheid

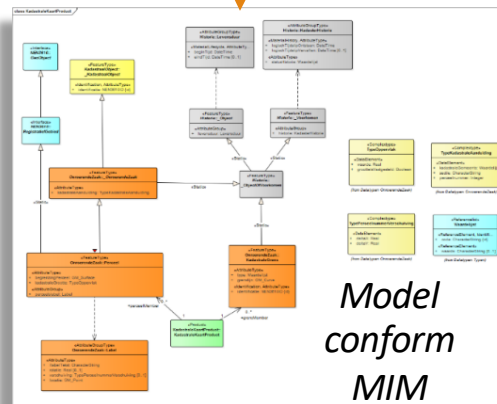
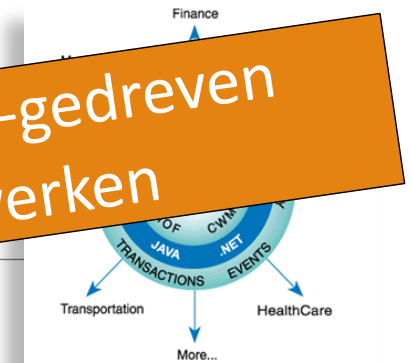
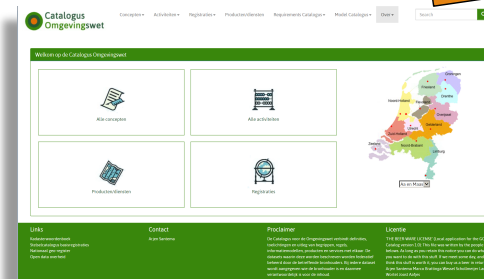


# Consistent doorvertalen = eenvoudig terug redeneren

Model-gedreven  
werken



## Begrippenkader



XML document

Service: XML bericht

API: JSON bericht

Database model

# de standaard: metamodel MIM

<https://www.geonovum.nl/geo-standaarden/metamodel-informatiemodellering>

---

Het metamodel bestaat uit

een verzameling van de bouwstenen c.q. de modelelementen  
die gebruikt (mogen) worden om een informatiemodel mee op te stellen.

Het is de **modelleertaal waarin een informatiemodel** is uitgedrukt.

## Doelen:

- **gezamenlijke taal en structuur** - leesbaarheid en eenduidigheid vergroten
- om het **koppelen** van diverse informatie/modellen te vereenvoudigen
- **tooling** en model-driven werkwijze, om koppelvlakken en producten te maken
- **kennis uit organisaties te bundelen** en bij elkaar te brengen



kadaster



...



1

# Metamodel wordt gebruikt voor

Begrippenkader

Conceptueel  
informatiemodel

Logisch  
informatiemodel

Fysiek/technisch  
datamodel

De onderwerpen van gesprek  
(universe of discourse),  
met onderlinge samenhang  
(en met andere begrippenkaders)

Voorbeeld:  
Bedrijfswoordenboek, juridisch  
begrippen in een wet zoals BAL

Bouwwerk, bouwvergunning,  
eigenaar, vervreemder, verkrijger,  
persoon

Implementatievrij, en vrije  
modelstructuur, naar behoefte.

Structuren van informatie in  
een informatie model  
(en samenhang met andere  
informatie modellen).

Voorbeeld:  
RGB 2.0, IMBAG 2.0, ...

Pand, openbare ruimte,  
object afbakening, natuurlijk  
persoon met BSN, niet-  
natuurlijk persoon met OIN,  
KvK nr

Implementatie vrij, geen:  
technische optimalisaties,  
geen historie, geen GML...

Hoe de **informatie gebruikt  
wordt bij de interactie tussen  
systemen**, tussen gebruikers, of  
tussen systemen en gebruikers  
en (veelal) precies geregistreerd  
wordt via inwinningsprocessen:

Het Interoperabiliteitsdomein

Voorbeeld: uitwisselingsmodel  
zoals IMKAD in NEN3610 (met  
ID), URI's, productmodel, intern  
model van de voorziening.

Implementatiekeuzes zoals  
INSPIRE compliant, GML 3.2.1.,  
Abstracte class BGT object, met  
historie en generieke attributen.  
Generiek tekstelement, Gebied.

Technische representatie van  
het logische model in een  
bepaalde **specifieke techniek**.

Voorbeeld; XSD, JSON  
schema,  
relationele database, tripple  
store, document store etc.

Diverse technische keuzes,  
t.b.v. performance, fysieke  
opslagstructuur, keuzes welke  
controles in de software en  
wat in de database, srs-  
transformaties etc...

## 2

# Metamodel toegelicht – de modelelementen waarmee je modelleert

### *MIM metaclass - kern*

Objecttype

Attribuutsoort

Gegevensgroep

Gegevensgroeptype

Generalisatie

Relatiesoort

Relatieklasse

Relatierol target

### *MIM metaclass - datatypeen*

Primitief datatype

Gestructureerd datatype

Data element

Union

Union element

Enumeratie

Enumeratiewaarde

Referentielijst

Referentie element

Codelist

### *MIM metaclass – overig*

Constraint

Relatierol (abstract)

Relatierol source

Externe koppeling

Domein

Extern

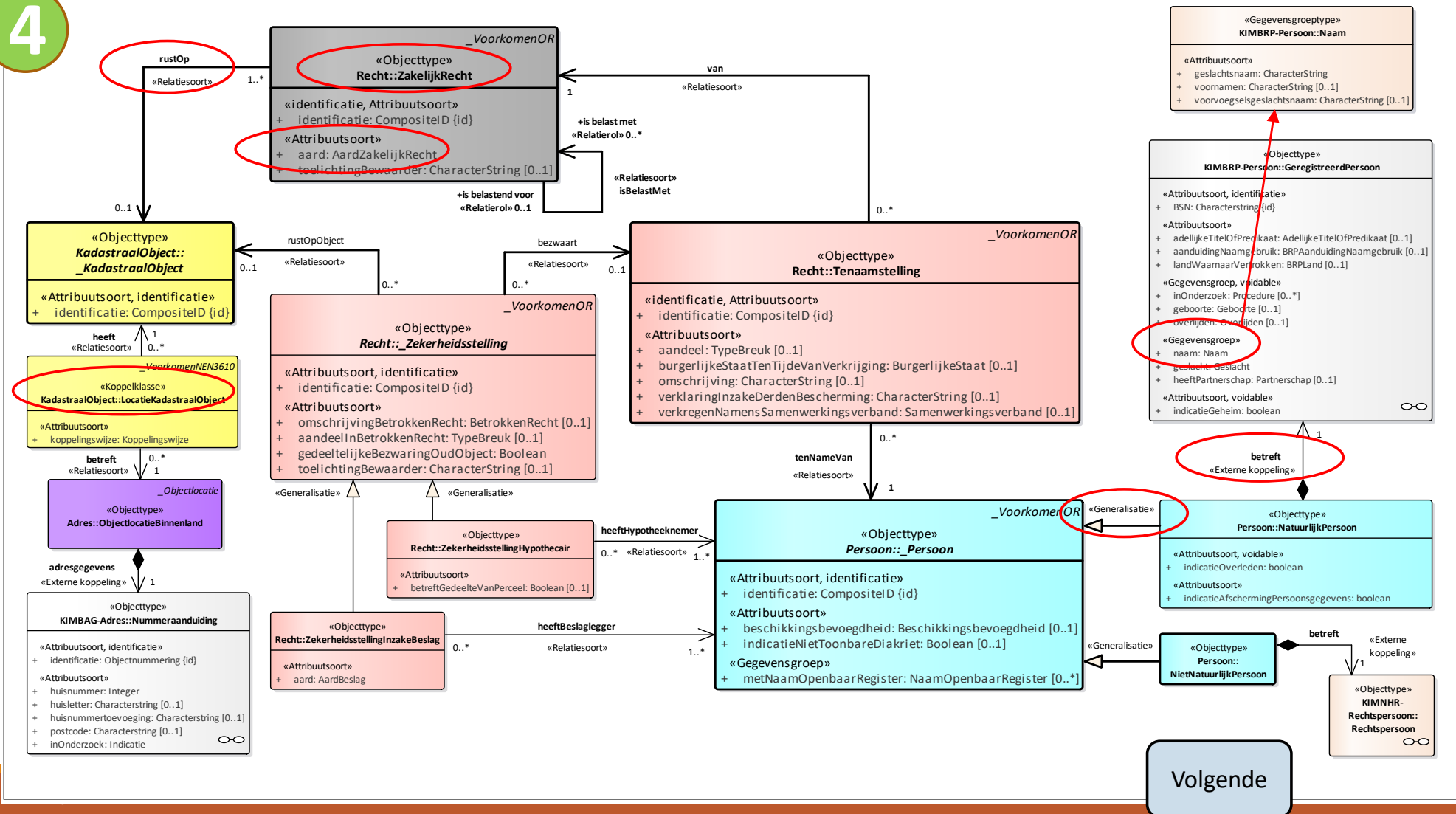
View

Merk op: niet alle hoeven van toepassing te zijn of gebruikt te worden.

## Metamodel – uitgedrukt in ...

Maar dit kan ook zijn: UML zoals in Enterprise Architect, of in bv. SHACL

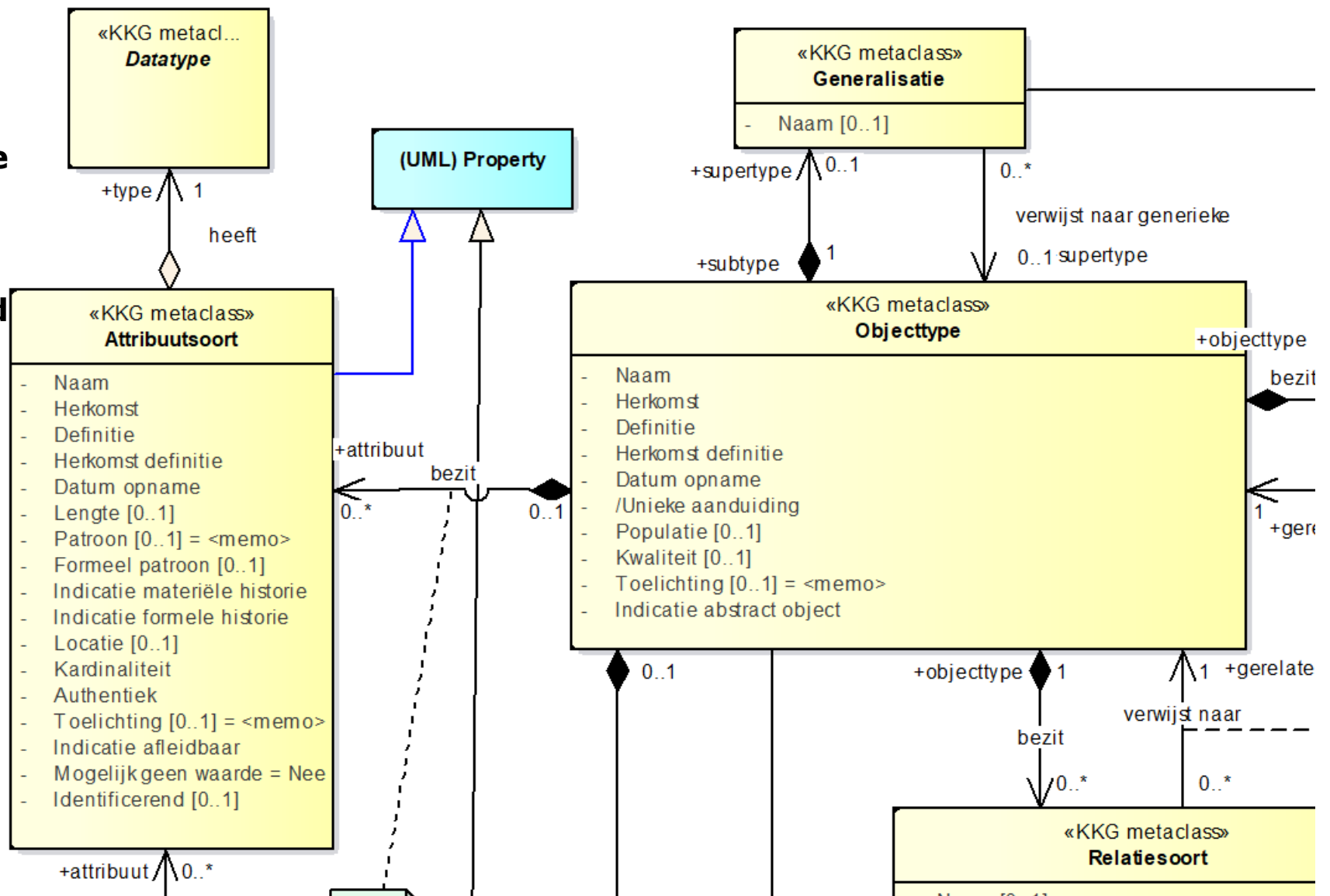
<b>MIM metaclass</b>	<b>Metaclass UML 2.5</b>	<b>Stereotype</b>
<b>Objecttype</b>	(UML) Class	«Objecttype»
<b>Attribuutsoort</b>	(UML) Property	«Attribuutsoort»
<b>Gegevensgroep</b>	(UML) Property	«Gegevensgroep»
<b>Gegevensgroeptype</b>	(UML) Class	«Gegevensgroeptype»
<b>Generalisatie</b>	(UML) Generalization	«Generalisatie»
<b>Relatiesoort</b>	(UML) Association	«Relatiesoort»
<b>Relatierol target</b>	(UML) Property	«Relatierol»
<b>Relatieklasse</b>	(UML) Association én (UML) Class	«Relatieklasse»



5

Beschrijvende kenmerken.

Zorgt voor eenduidigheid



6

## Metamodel – uitgedrukt in Linked Data vocabulaire

---

Work in progress



7

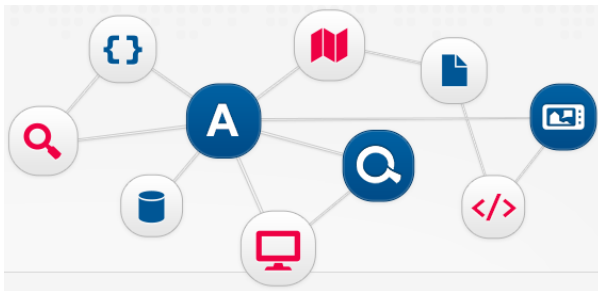
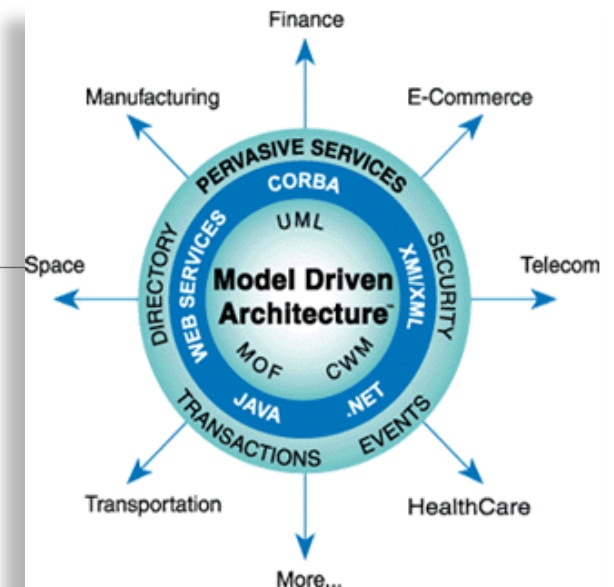
# Automatisch genereren

EA: UML profielen voor modelleers

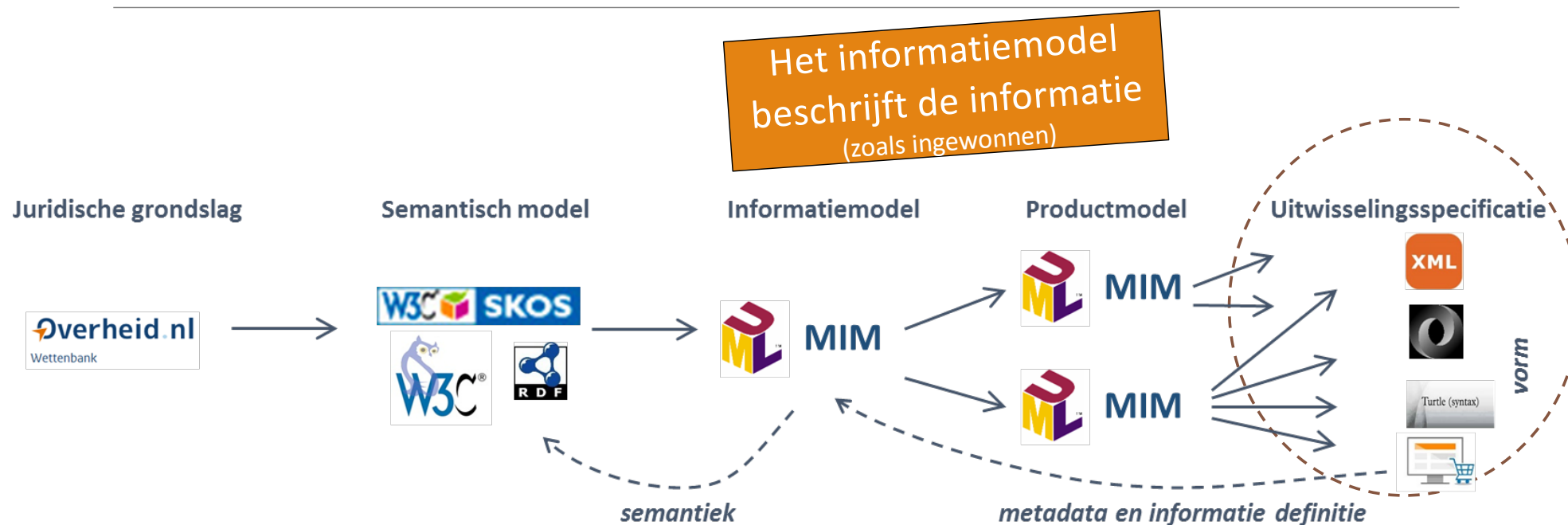
Tooling: bv. IMvertor: [www.imvertor.org](http://www.imvertor.org)

Generatie van afgeleide producten:

- XML schema, RDF, andere
- Objectcatalogus in verschillende formaten (HTML, ... )



# MIM maakt vaste vertalingen naar uitwisselformaten mogelijk

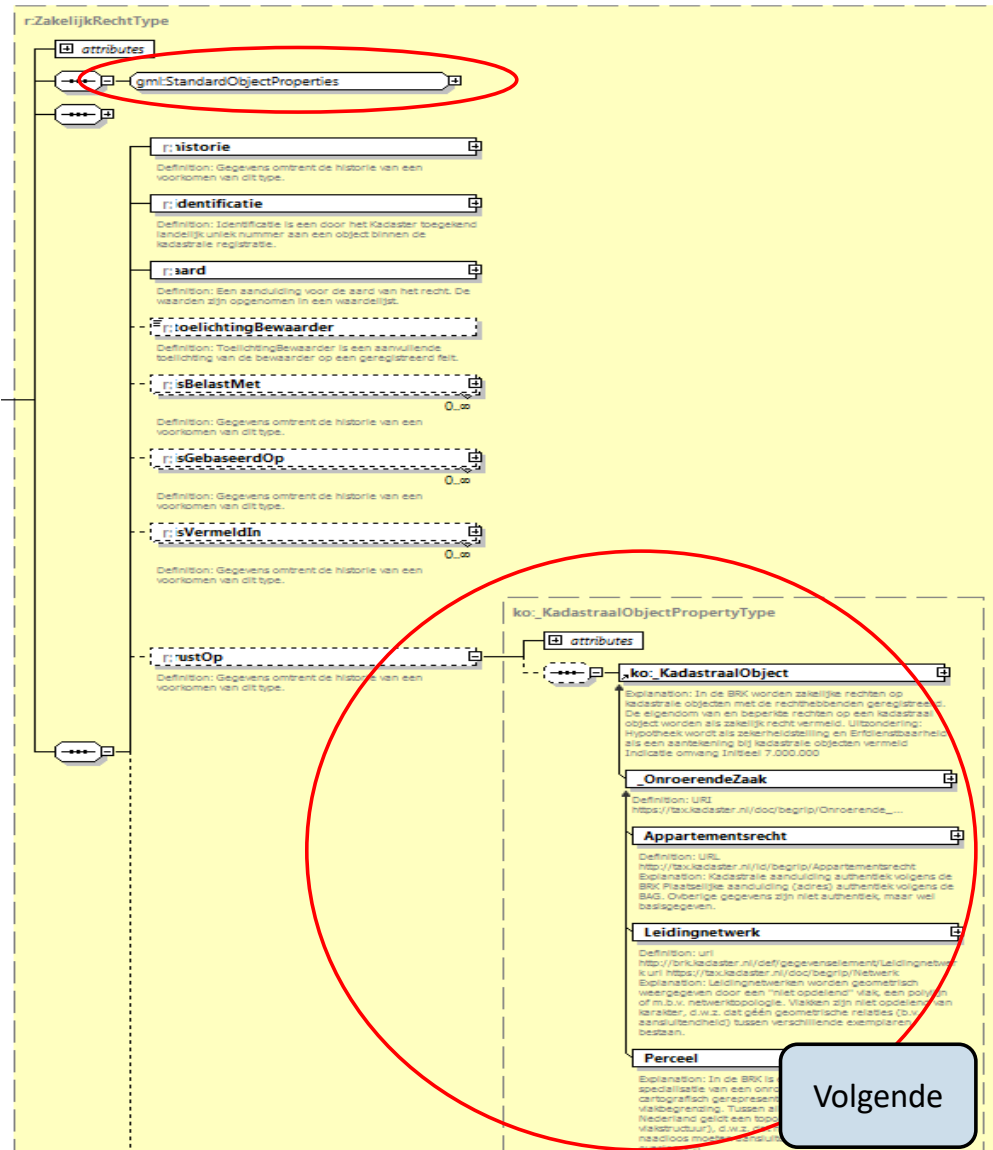
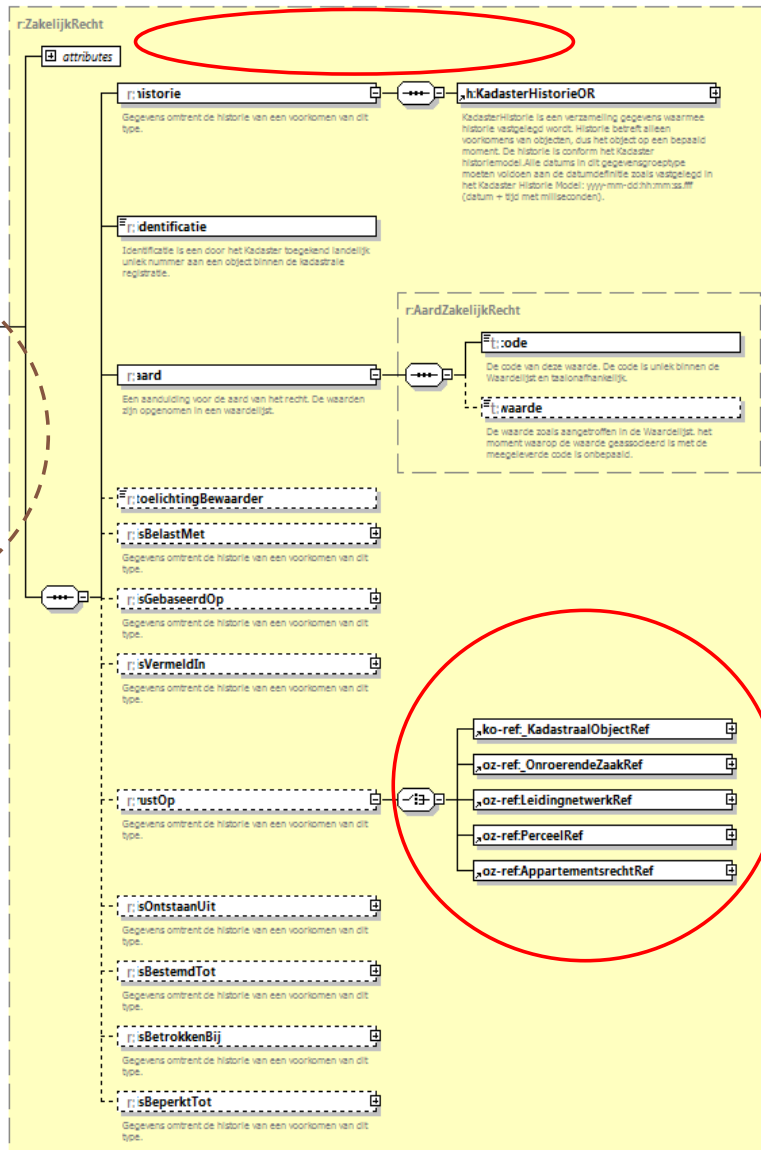




# IM in MIM, standaard vertaling naar XSD t.b.v. ML of GML

## ZakelijkRecht

In afwijking van de definitie van het juridisch begrip Zakelijk recht is in de BRK een zakelijk recht een recht dat samenhangt met een kadastraal object. Bovendien worden de volgende rechten afwijkend geregistreerd: het Recht van hypotheek (samen met een beslag) vermeld als een Zakenrecht bij een kadastraal object. \* Een erfloosbaarheidsvermeld als een Aantekening bij kadastrale objecten. \* Een appartementsrecht opgenomen als een kadastraal object. Een zakelijk recht in de BRK betreft: eigendom\* Beperkte rechten: Recht van erfpacht, Recht van opstal, Recht van vruchtgebruik, Recht van gebruik en bewoning), \* rechten van voor 1992 (volkrecht en zakelijk recht als bedoeld in de Besluitvormingswet privaatrecht\*) oud vaderlandse rechten (bijv. vassalrechten, grondrente enz\*) Zakelijk opterecht bij luchtbeurten



Volgende

# IM in MIM: standaard vertaling naar JSON schema

MIM modellement	JSON schema	Waar opnemen en toelichting/voorbeeld.
<<Objecttype>>	NaamObjecttype" : { "title" : "<naam in IM>", "type" : "object",	In definitions": { Bv. "Pand": { "title": "Pand", "type" : "object" }
<<Attribuutsoort>>	"naamAttribuutsoort" : { "type" : <primitief datatype> } OF "\$ref": "#/definitions/<defined datatype>"	Binnen "properties" : { Bv. "bouwjaar" : { "type": "string" } Bv. "waarde": { "\$ref": "#/definitions/WaardeEenheid" }
<<Relatiesoort>> (0..1 of 1..1)	"naamRelatiesoort": { "\$ref": "#/definitions/<target objecttype>" },	Binnen "properties" : { Bv. "ligtIn": { "\$ref": "#/definitions/Woonplaats" }
<<Relatiesoort>> (0..* of 1..*)	"naamRelatiesoort": { "type": "array", "items": { "\$ref": "#/definitions/<target objecttype>" }, minItems : <integer> } ,	Binnen "properties" : { Bv. "naastgelegen": { "type": "array", "items": { "\$ref": "#/definitions/Pand", minItems : 0 } <i>Kies de naam van relatiesoort, of van relatierol, niet beide.</i>
<<Gestructureerd datatype>>	"NaamDatatype" : { "type": "object" , "properties": "<data element1>" : { "type" : <datatype> }, "<data element2>" : { \$ref : <datatype> } }	Opnemen bij "definitions": { Bv. "WaardeEenheid": { "type": "object", "properties": { "waarde": { "type": "number" }, "eenheid": { "type": "string" } } }
Naam in IM	"definitions": { "JuridischeRegel": { "title": "Juridische regel"	Zodat een API ook kan kiezen om een andere naam te gebruiken en de title kan behouden als originele naam.
Definitie	" <u>description</u> " : "De tekst van definitie"	Bij objecttype, na "type": "object", "description": "..." Bij attribuutsoort, na "properties": { "naamAttribuut": { "description": "" } Bij relatiesoort, direct na "\$ref".
Kardinaliteit	1: staat wel in required. 0: niet in required. *: meer: gebruik array (minItems 1 als required, 0 als niet).	Let op, required én minItems 0 mag niet.

[Home](#) / [Geo-standaarden](#) / Metamodel Informatiemodellering...

## Metamodel Informatiemodellering (MIM)

Om alle informatiemodellen in Nederland nog beter op elkaar aan te laten sluiten hebben VNG Realisatie, Kadaster en Geonovum gezamenlijk een metamodel ontwikkeld voor informatiemodellering. Hierin komt de modelleringskennis van deze drie organisaties samen.

→ [Bekijk alle Metamodel Informatiemodellering \(MIM\) standaarden](#)

Informatie is een motor onder het functioneren van de overheid in Nederland. In het kader van onderlinge samenwerking, wisselen overheidsorganisaties veel informatie uit. Daarbij is het van groot belang dat we hetzelfde verstaan onder de gegevens die we gebruiken en dat we gemeenschappelijke afspraken maken over hoe we gegevens beschrijven en uitwisselen.



*Informatie in gezamenlijkheid*

# MIM verbindt en ontzorgt de informatieoverheid

**MIM helpdesk:** [mim@geonovum.nl](mailto:mim@geonovum.nl)

**MIM discussieforum:**

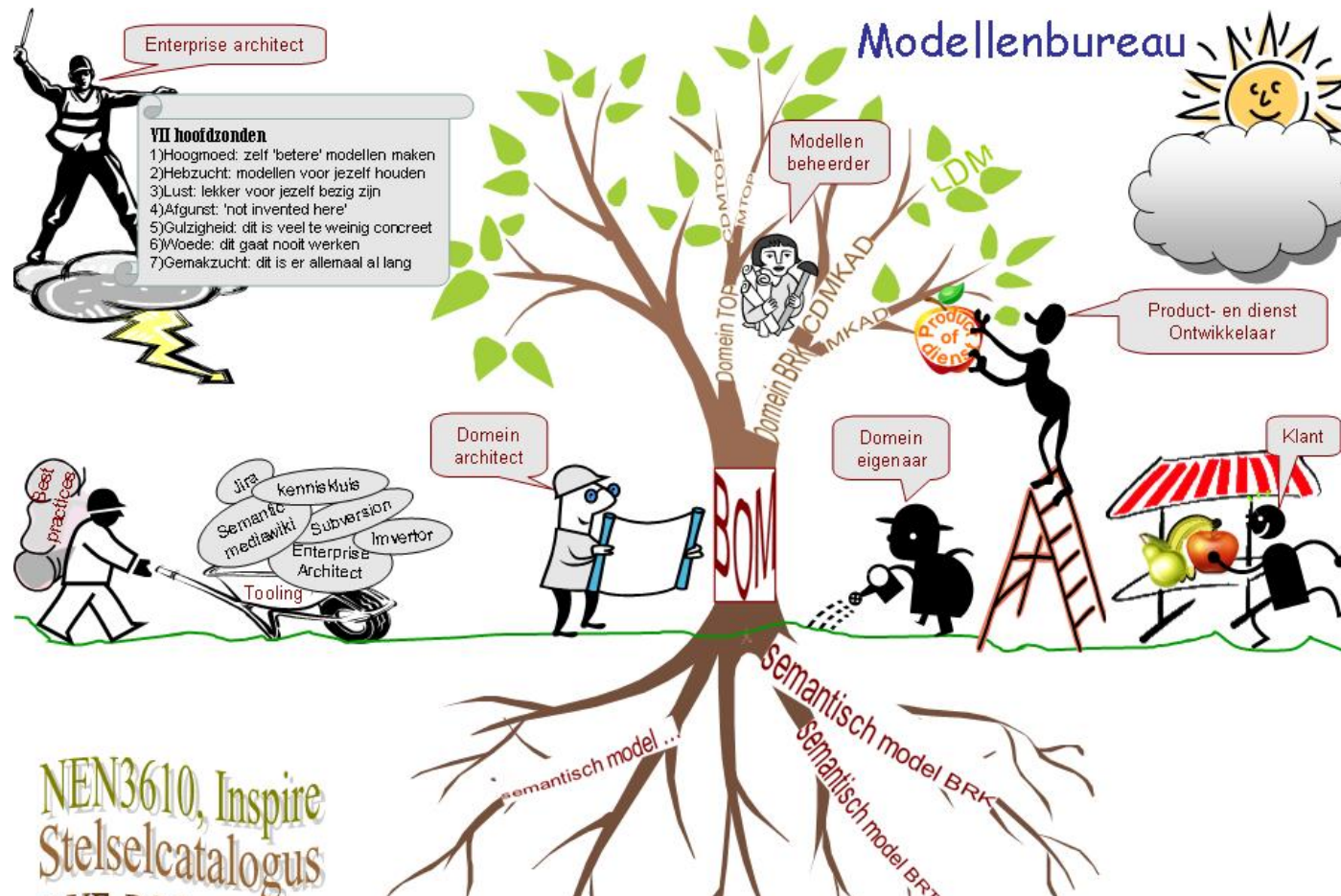
**MIM documentatie:**  
<https://docs.geostandaarden.nl/mim/mim10/>

# INZOOMEN? MAAK UW KEUZE

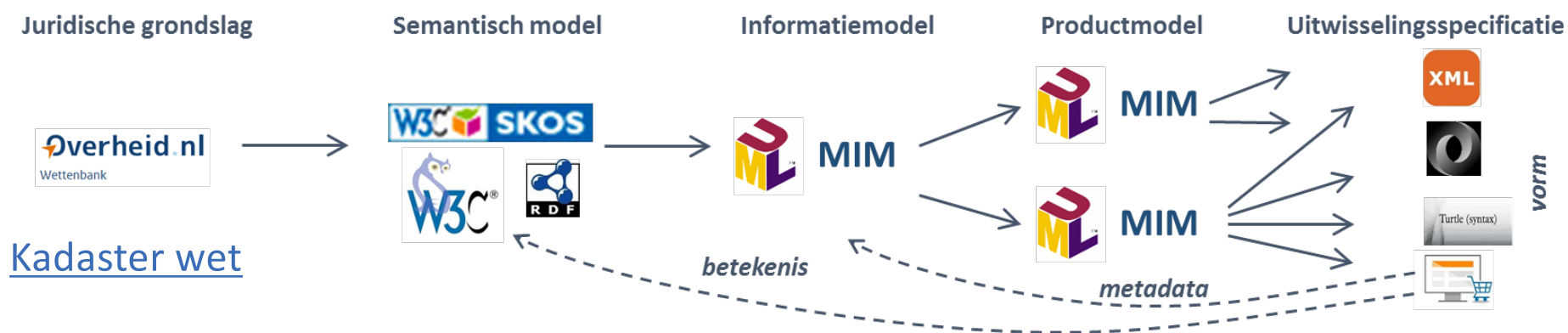
Kadaster werkwijze:

1. Community building en organisatie met een **modellenbureau**
2. Een stappenplan voor het **model-gedreven werken** vanuit het informatiemodel en producten naar technische schema's en API's
3. TOOL DEMO: XSD en JSON schema **genereren** met Enterprise Architect en Invertor
4. Discussie en/of algemene vragenronde

# 1. COMMUNITY BUILDING EN ORGANISATIE



# 1. IN SAMENHANG BEHEREN EN MODEL-GEDREVEN WERKEN



De stappen hebben een sterke onderlinge samenhang en worden als geheel beheerd.

Dit leidt tot concrete voordelen, zoals:

- Domein deskundigen (juridisch en semantisch) geven kennis als kleine blokjes door;
- Productmodellen zijn flexibel samen te stellen uit deze blokjes (en in overeenstemming);
- Ontwikkelaars kunnen zich richten op het snel realiseren van technische formaten.

## 1. BEHEREN VAN HET GEHEEL

- Semantiek en begrippen <https://brk.basisregistraties.overheid.nl/begrippenlijst>
- Informatiemodel (intern en extern)
- Productmodellen
- Technische schema's
- API's en services

## 2. PRODUCTEN EN DATA GERICHT WERKEN - *ORIËNTEREN*

### **Product gericht werken**

- welke informatie heeft mijn proces of taak nodig?

### **Data gericht werken**

- waar vind ik de informatie, uit welke (officiële) bron is deze op te vragen?
- wat is de betekenis van de informatie uit deze bron?
- levert deze bron deze data, zo ja hoe?

### ***... en wat als het product of de data er nog niet is?***

- wil ik een product gaan aanvragen om deze data wel geleverd te kunnen krijgen?
- of moet/wil ik deze data zelf gaan inwinnen en registreren?
- wil ik enkelvoudige data objecten / resources zelf combineren of een echt product? En hoe zit het dan met de exacte betekenis en met complexe situaties en onderliggende bedrijfsregels?



## 2. PRODUCTEN EN DATA GERICHT WERKEN: *TRENDS VOOR LEVEREN DATA*

### **Kant en klare producten die goed aansluiten bij de behoefte**

- Organisaties gaan in gesprek met de beheerder van de bron, en komen samen tot een product met de benodigde informatie;
- Bedrijfsregels zijn verwerkt in het product, zo eenvoudig als mogelijk;
- Liever geen losse resources om te combineren en voorgaande zelf uit te zoeken\*.

### **De informatie is helder en begrijpelijk**

- Betekenis en structuur is goed beschreven en is (vrijwel) altijd hetzelfde;
- De data is conform de betekenis (eerst controleren, dan pas leveren);
- Betekenis wordt integraal meegeleverd bij de informatie.

### **Het technisch formaat sluit aan bij bestaande/beoogde IT oplossingen van de klant**

- XML en JSON zijn het meest gewenst, maar klanten willen ook bv. vector tiles of linked data

## 2. MODEL-GEDREVEN WERKEN: *INFORMATIE STANDAARDEN*

### **Stap 1: Beschrijf eerst de semantiek van het domein**

- Begrippen/woordenboek: onderling samenhang en met andere domeinen (bv. SKOS/OWL).

### **Stap 2: Maak een informatiemodel van deze informatie, met betekenis en structuur**

- Dit model geeft aan wat de bron bevat, ingewonnen in deze betekenis en structuur;
- Het informatiemodel is techniek onafhankelijk (niet toegespitst op één techniek), pas (technologie onafhankelijke) standaarden toe: metamodel voor informatiemodellen\*
- Maak je informatiemodel geschikt voor uitwisseling, door uitwisselingsstandaarden toe te passen. In geval van de BRK / IMKAD is dit bijvoorbeeld de NEN3610 standaard.
  - Als het model voor meerdere koppelvlakken of ketens geschikt gemaakt moet worden, en er (op informatiemodel niveau) sprake is van verschillende toe te passen uitwisselingsstandaarden, dan kunnen er meerdere varianten ontstaan.

## 2. MODEL-GEDREVEN UITWERKEN: *TECHNISCHE STANDAARDEN*

### **Stap 3: Product maken - leidt een product (of kleine resources) af van het informatiemodel**

- Product design: de gebruiker heeft bepaalde informatie nodig in zijn proces → product.
- Service/API design: de gebruiker wil in een aantal stappen informatie opvragen – interacties met het product. Het product wordt geleverd met een service, of fijnmazig interacties ondersteunen → *API*.

### **Stap 4: Maak *technische schema's* voor producten (en/of kleine resources) per formaat**

- XML/*XSD*/Digikoppeling;
- GML/*XSD*/OGC;
- JSON/*JSON-schema*/API strategie;
- TTL/*RDF-S*/HTTP&URI strategie;
- Specificatie is nodig om het informatiemodel uit te drukken per technische standaard\*  
(een vaste vertaling, zonder betekenis te verliezen, of structuren onnavolgbaar te wijzigen en niet per product of dienst helemaal verschillend)

## 2. PRODUCT DESIGN EN API DESIGN: JSON VIA EEN API

De gebruiker heeft bepaalde informatie nodig in zijn proces. Een product bevat deze informatie. Welke informatie in het product komt, heet *product design*.

De gebruiker wil in een aantal processtappen deze informatie gebruiken/opvragen → interacties met product. De API zal deze fijnmazige interacties ondersteunen. Dit optimaal inrichten van de interacties, inclusief performance optimalisatie, heet *API design*.

In 1 “product” API zitten, om goed door het product kunnen navigeren, daarom apart gedefinieerde interacties / toegangspaden / *API calls*, om delen van de informatie op te vragen, via de API:

- De primaire toegang is de identificatie van een object, of de meest gebruikte zoekvragen;
- Bij elk antwoord zitten veelgebruikte **vervolgvragen** (incl. specifieke kennis) al als ingebakken links in het antwoord, waar de gebruiker, of het systeem, alleen maar op hoeft door te klikken;
- Nieuwe zoekpaden of links kunnen veelal back-wards compatible toegevoegd worden;  
.. hoewel, let op: nieuwe manieren om door het product te navigeren kunnen een verrassing worden voor de back-end.

### 3. DEMO